



Performance Comparison of Supervised and Reinforcement Learning Approaches for Separating Entanglements in a Bin-Picking Application

Marius Moosmann¹(✉), Manuel Kaiser¹, Johannes Rosport¹, Felix Spenrath¹,
Werner Kraus¹, Richard Bormann¹, and Marco F. Huber²

¹ Department Robot and Assistive Systems, Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Nobelstraße 12, 70569 Stuttgart, Germany
marius.moosmann@ipa.fraunhofer.de

² Department Cyber Cognitive Intelligence (CCI), Fraunhofer IPA and Institute of Industrial Manufacturing and Management IFF, University of Stuttgart, Stuttgart, Germany

Abstract. Machine Learning helps to separate entanglements in Bin-Picking Applications. The goal is to create a system that finds a path to separate an entanglement, starting from a single visual input. To realize such a system both supervised and reinforcement learning methods can be implemented. For both of these approaches we set up a motion model and the remaining properties of the real robot cell are implemented in a simulation scene. While the simulation scene can be used to create training data for the supervised learning approach, it is also the learning environment for the reinforcement learning model. Therefore, there are similar premises for comparing the two models. What needs to be investigated is which of the two methods separates the most entanglements and offers the least setup effort. The setup effort in general and the performance are examined for both approaches in simulation and also in real-world experiments. The reinforcement learning model outperforms both of the supervised learning models in the setup effort and the separation rate by over 15 percent points.

Keywords: Bin-Picking · Machine Learning · Reinforcement Learning · Supervised Learning · Entanglement Separation

1 Introduction

The ability to automatically pick chaotically stored workpieces from bins creates many new opportunities in productions. While some workpiece geometries can be picked in a robust manner, there are a lot of workpieces which are prone to entangle. Therefore an automated process that is supposed to consistently pick single workpieces needs to have the ability to detect and separate entanglements. Since the detection of entanglements has been handled in previous work [1, 2], this paper mainly focuses on the separation of these entanglements. For this

purpose a new convolutional neural network (CNN) architecture for a supervised learning approach has been developed and tested alongside two already existing supervised learning [3] and reinforcement learning [4] approaches.

The main contributions of this paper are:

- the introduction of the updated supervised learning architecture
- the introduction of a threshold evaluation to deny gripping points for impossible separations
- real-world entanglement separation experiments
- comparison of supervised and reinforcement learning approaches

2 State of the Art

In motion-planning for bin picking applications multiple different approaches have emerged recently. Ellikide et al. [5] and Iversen et al. [6] prioritize finding a motion path, which avoids collisions with the environment. In the case of separating entangled harnesses Zhang et al. [7] uses a set of eight possible motion schemes with increasing complexity. Matsumura et al. present a model-free entanglement detection approach, but without separation strategies [8]. Leao et al. calculate the robot trajectory based on the size of the workpiece and move the robot on its x - y -plane [9]. Moosmann et al. [3] proposed a motion model in the shape of a hemisphere consisting of 25 points, which is based around the center of the entangled workpiece. The amount of hemisphere points have later been reduced to 17 points [4]. Each of the points has a specific translational and rotational offset that gets added to the original workpiece position. The workpiece is moved to the point, which has the highest probability to separate the entanglement. To calculate these probabilities a supervised learning [3] and a reinforcement [4] learning approach has been developed. Additionally two more hemispheres are created centering around the selected path point of the preceding hemisphere. Once all three path points have been passed the workpiece is lifted up. Figure 1 displays an example for a separation in the simulation environment.

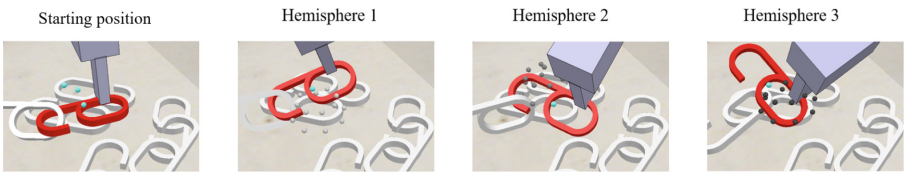


Fig. 1. Example for an entanglement from the simulation with the corresponding separation path generated by the Entanglement Separation Network

3 New Supervised Learning Entanglement Separation Method

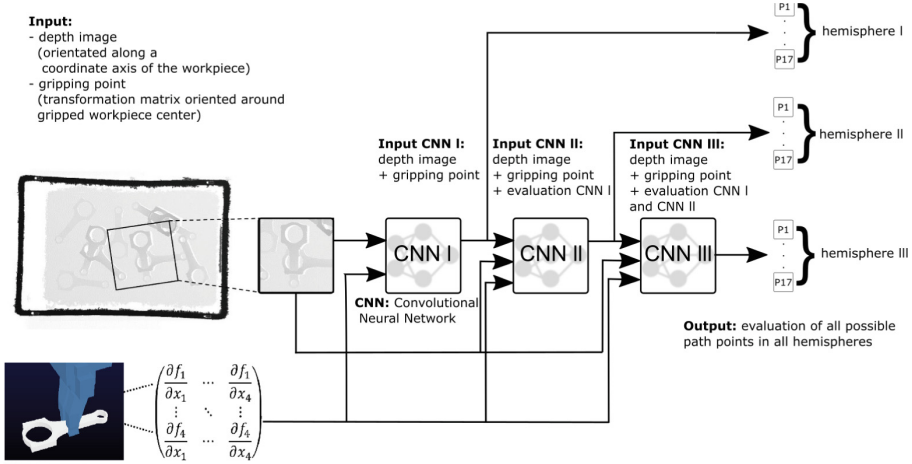


Fig. 2. Supervised Learning A: Serial connection of CNNs presented in [3].

In [3] a supervised learning method was presented, which uses a serial connection of three convolutional neural networks to predict the optimal trajectory for entanglement separation as shown in Fig. 2. This approach has been unified into a single network within this work in order to simplify the usage and reduce training time.

3.1 Data Generation

The training data is generated using the simulation environment CoppeliaSim¹. In the simulation scene several objects such as the workpieces and bins in multiple sizes are integrated. All are based on a CAD-model with real-world proportions. A simulation cycle starts by filling the bin with a random amount of background workpieces, which vary between 0 and 20. After that a random entanglement is selected from of previously generated entangled workpiece poses and placed with a random x and y offset in the bin. To make sure the entanglement is still valid after being placed in the bin, the entangled workpiece is lifted up and it is checked how many workpieces are located above the bin. As soon as the conditions for a valid entanglement are met, a set of possible gripping points is checked in the simulation. For every gripping point that does not collide with the bin or the surrounding workpieces, a simulation cycle will start.

¹ <https://coppeliarobotics.com/>.

After a valid gripping point has been chosen, the path points of the first hemisphere are checked. A separation path is considered successful, if neither the gripper nor the workpiece collide with the bin and the entanglement has been separated. The second and third hemisphere is created for the best point of the preceding hemisphere. In the case of multiple or none successful separations, the path which caused the least movement for the surrounding workpieces is selected as the center for the next hemisphere. One simulation cycle is finished as soon as the 17 path points of each hemisphere have been checked. The separation motion model is presented in Fig. 3. On the left, the distribution of the 17 possible path points on the hemisphere, on the right, a possible trajectory with three hemispheres is shown.

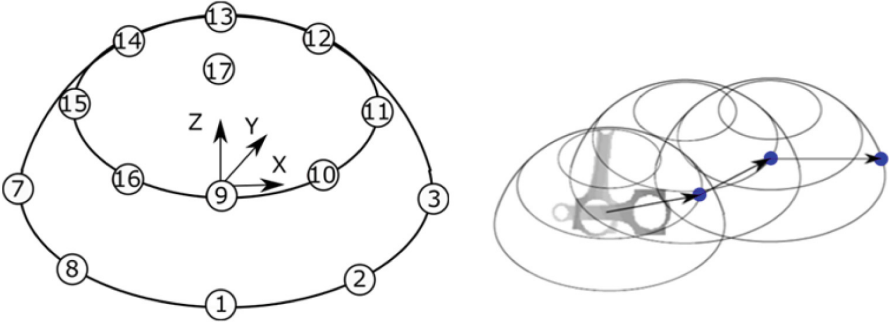


Fig. 3. Separation motion model - left: 17 possible path points of a hemisphere; right: possible trajectory with three hemispheres

The depth images, which are taken for every cycle have a size of 128×128 pixels. Before training the network with these images they are transformed using transfer learning methods as shown in [1]. This is necessary to minimize the differences between simulation and real-world data. For domain adaptation, we use CycleGAN [12] to generate real-looking depth maps from simulation. To get more realistic sensor data, we add different domain randomisation factors to the simulation depth map, for example Gaussian noise, translational and rotational offsets and brightness adaption [13, 14]. We also use inpainting techniques [15].

3.2 Network Architecture

Since for the new architecture all three networks as presented in Fig. 2 have been merged into a single one, the information about the previously selected actions and the current hemisphere index needs to be transferred in a consistent manner. Therefore a matrix with 2×17 values is used as an additional input. The first column contains the value “one” at the index of the selected action in the first hemisphere and the value “zero” for the remaining indices. For the second column the value “one” is contained at the chosen action for the second

hemisphere. If the respective hemisphere has not been evaluated yet the vector contains 17 times the value “zero”. The gripping point is represented by a 4×4 transformation matrix relative to the workpiece center. This means a simulation cycle as described in Sect. 3.1 creates three training labels with the same gripping point and depth image, but a varying previous action vector. The output of the network is the probability to solve the entanglement with each of the 17 hemisphere points. In Fig. 4 the complete architecture is summarized.

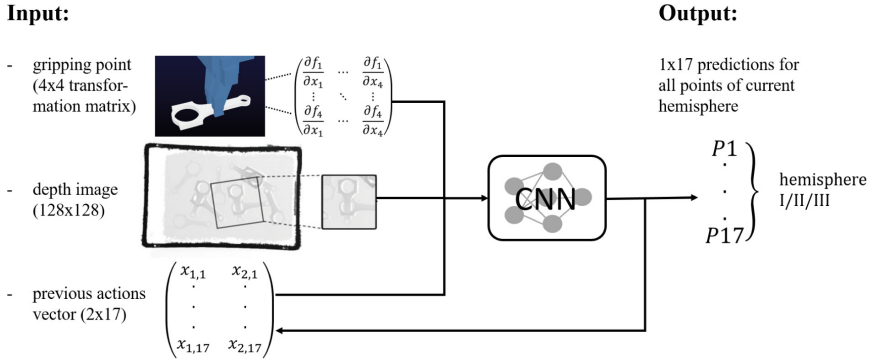


Fig. 4. Input and Output of the Entanglement Separation Network

For the separation task a DenseNet [16] architecture with 4 Denseblocks with depths of 6, 12, 24 and 12 layers is implemented.

3.3 Training

For all of the workpieces, which will be examined in this paper, around 20,000 data samples have been generated according to the procedure described in Sect. 3.1. The network has been trained using a Nvidia GeForce GTX 1080 Ti graphics card with a batch size of 128 for 80 epochs. To avoid overfitting, a dropout with a dropout rate of 35% has been used.

3.4 Threshold Evaluation

With the current motion model most entanglements have multiple different paths, which lead to separation. However some entanglements are impossible to separate. One reoccurring problem is the gripper blocking the path of the entangled workpiece as shown in Fig. 5 (a). Aside from that some entanglements are significantly easier to separate if another workpiece, which is part of the same entanglement, is gripped. To address these problems we implemented the ability to deny the gripping of a workpiece if the average value of the 17 predictions of the first hemisphere is below a threshold. To find the best threshold, the validation data of our training set has been examined for every workpiece on multiple thresholds between 0 and 0.1. The threshold with the highest accuracy has shown to be 0.022 as depicted in Fig. 5 (b).

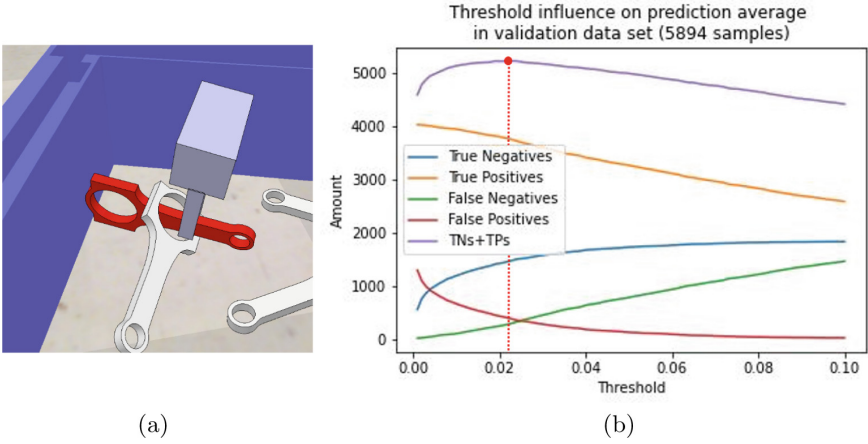


Fig. 5. (a) Example of an unsolvable gripping point (b) Threshold influence on prediction accuracy of the connecting rod with maximum at 0.022

4 Comparison

4.1 Setup Effort

For the supervised learning approach the amount of training data is a substantial factor for training success. Therefore it is necessary to generate a large amount of training data, which is the most time consuming part. To generate data from one simulation cycle as described in Sect. 3.1, a Lenovo Thinkpad with an Intel(R) Core(TM) i7-10750H CPU with 2.60 GHz processor and 16 GB RAM takes about 4.5 min on average. Accordingly generating 20,000 data samples would take 1,500 h. However since simulations can run on multiple processor cores simultaneously, the time to generate this data can be divided by the amount of cores on the respective system, reducing data generation time significantly. To keep the conditions for this comparison equal this aspect will be ignored. The training time of the updated supervised approach is 0.5 h lower than the previous approach.

In one episode of the reinforcement learning training a single separation path in the simulation environment is tried. Therefore the reinforcement learning approach takes up significantly less time per episode, compared to a cycle of the supervised learning data generation. Aside from that, the reinforcement learning training does not need any previously generated data and uses an Epsilon-Greedy strategy to explore the environment within the first 1,000 episodes. To achieve a sufficiently trained network about 30,000 episodes are necessary, which takes around 175 h. In Table 1 time consumption is summarized concluding that the reinforcement learning take up clearly less time.

Table 1. Time consumption comparison of Supervised and Reinforcement Learning approach, Supervised Learning A is referring to the new merged architecture and Supervised Learning B is referring to the old serial connection architecture

Approach	Required Sim Data for Training	Setup Time	Training Time	Total Time
Supervised Learning A	$\approx 20,000$	$\approx 1,500$ h	≈ 1 h	$\approx 1,500$ h
Supervised Learning B	$\approx 20,000$	$\approx 1,500$ h	≈ 1.5 h	$\approx 1,500$ h
Reinforcement Learning	0	0	≈ 175 h	≈ 175 h

4.2 Real-World Performance Comparison

To evaluate the performance of the networks, real-world experiments have been carried out. In order to see how the separation rate differs for a variety of workpiece geometries, u-bolts, connecting rods and hooks have been tried. The methods under consideration for the following comparison are the reinforcement learning approach introduced in [4], the supervised learning approach introduced in [3] and the new supervised approach introduced in this paper. For every combination of those workpieces and machine learning methods, the entanglement separation success rate for 200 workpieces has been determined as depicted in Table 2. All tests up to the last row do not involve the threshold, which has been introduced in Sect. 3.4. Comparing the results of the different supervised learning networks, the only minimal differences for all workpieces range between 0.5 and 1.5% points. However comparing the reinforcement learning network with the supervised learning approach the separation rates differ within a greater range. Here the u-bolt workpiece geometry shows the best separation rate with 98%, being 15.5% points better than the new supervised approach. For the hook and the connecting rod a slightly better separation rate can be achieved.

To evaluate the effectiveness of the threshold, another test series with the connecting rod has been carried out, in which predictions below the threshold were denied. This workpiece has been prioritised for the threshold evaluation, because it is more prone to impossible gripping points as demonstrated in Fig. 5 (a). The improvement of these test cases is shown in the last row of Table 2. Here an additional improvement between 4 and 5 percent points for both supervised learning approaches and the reinforcement learning approach is visible.

Table 2. Separation test comparison for all combinations of machine learning methods and workpieces (with 200 entanglements per combination)

Object	Approach	Entanglements separated	Separation Rate
U-bolt	Supervised Learning A	165	82.5%
	Supervised Learning B	162	81.0%
	Reinforcement Learning	196	98.0%
Hook	Supervised Learning A	92	46.0%
	Supervised Learning B	91	45.5%
	Reinforcement Learning	101	50.5%
Connecting Rod	Supervised Learning A	174	87.0%
	Supervised Learning B	176	88.0%
	Reinforcement Learning	176	88.0%
Connecting Rod With Threshold	Supervised Learning A	183	91.5%
	Supervised Learning B	184	92.0%
	Reinforcement Learning	186	93.0%

4.3 Integration into a Bin Picking System

The Bin-Picking Application, in which the separation strategies are integrated in, is able to detect and localize workpieces using a point cloud [10]. Furthermore to acquire a suitable gripping solution a heuristic search is used [11]. From the point cloud depth images are extracted and used as input for the entanglement detection. In case the workpiece is recognized as entangled a request for a separation path will be sent to the entanglement separation [1].

5 Conclusion and Future Work

In this paper it has been shown that the serial connection, consisting of three networks, which was previously implemented, can be reduced to a single network. This reduces training and load up time of the weights from the neural networks, without compromising in terms of performance. Additionally the real-world tests have shown that the reinforcement learning model achieves a 15% points higher separation rate, while having a lower setup effort. Furthermore, we introduced a threshold evaluation to deny gripping points, on the basis of which the separation of entanglements is impossible. This evaluation is also done in real-world experiments. In future work we will try to improve the reinforcement learning approach with any additional rotations and simplify the pipeline to teach in new workpiece geometries. Furthermore we will try to reduce training time with meta-learning methods.

Acknowledgement. This work was partially supported by the German Federal Ministry of Education and Research (Deep Picking - Grant No. 01IS20005C) and the Ministry of Economic Affairs of the state Baden-Württemberg (Center for Cognitive Robotics - Grant No. 017-180069 and Center for Cyber Cognitive Intelligence (CCI) - Grant No. 017-192996).

References

1. Moosmann, M., et al.: Transfer learning for machine learning-based detection and separation of entanglements in bin-picking applications. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2022)
2. Moosmann, M., et al.: Increasing the robustness of random bin picking by avoiding grasps of entangled workpieces. In: CIRP Conference on Manufacturing Systems (CIRP CMS) (2020)
3. Moosmann, M., et al.: Using deep neural networks to separate entangled workpieces in random bin picking. In: Procedia CIRP (2020)
4. Moosmann, M., et al.: Separating entangled workpieces in random bin picking using deep reinforcement learning. In: CIRP Conference on Manufacturing Systems (CIRP CMS) (2021)
5. Ellekilde, L.P., Petersen, H.G.: Motion planning efficient trajectories for industrial bin-picking. In: IJRR, pp. 991-1004 (2013)
6. Iversen, T.F., Ellekilde, L.P.: Benchmarking motion planning algorithms for bin-picking applications. *Ind. Robot.* **44**(2), 189–197 (2017)
7. Zhang, X., Domae, Y., Wan, W., Harada, K.: Learning a sequential policy of efficient actions for tangled-prone parts in robotic bin picking. In: arXiv (2021)
8. Matsumura, R., Harada, K., Domae, Y., Wan, W.: Learning based robotic bin-picking for potentially tangled objects. In: IEEE/RSJ IROS (2019)
9. Leao, G., Costa, C.M., Sousa, A., Veiga, G.: Detecting and solving tube entanglement in bin picking operations. *Appl. Sci.* **10**, 3390 (2020)
10. Palzkill, M.: Heuristic method for object pose detection using point clouds for automated feeding systems (2014)
11. Spenrath, F., Spiller, A., Verl, A.: Gripping point determination and collision prevention in a bin-picking application. In: 7th German Conference on Robotics (ROBOTIK) (2012)
12. Zhu, J.-Y., Park, T., Isola, P., Efros, A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision (ICCV) (2017)
13. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real-world. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017)
14. Borrego, J., Dehban, A., Figueiredo, R., Moreno, P., Bernardino, A., Santos-Victor, J.: Applying domain randomization to synthetic data for object category detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

15. Telea, A.: An image inpainting technique based on the fast marching method. In: Journal of Graphics Tools (2004)
16. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K. Q.: Densely connected convolutional networks. In: IEEE CVPR (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

